

Avatar

A sophisticated and flexible avatar component designed to display user profiles, status indicators, and visual representations with support for images, text labels, badges, and animated gradient borders. Perfect for user interfaces, chat applications, social platforms, and any system requiring user identification and status display.

How to use

Basic Usage

The most basic implementation with default settings and image support.

```
import { AavaAvatarsComponent } from "@aava/play-core";
```

Avatar Sizes

Seven distinct sizes to suit different design requirements and use cases.

```
<aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="User avatar"
/>
</aava-avatars>
```

Available Sizes

- xxs (Ultra Small) : Very compact size for extremely dense layouts
- xs (Extra Small) : Compact size for dense layouts and lists
- sm (Small) : Small size for compact interfaces
- md (Medium) : Standard size for most use cases
- lg (Large) : Large size for prominent content (default)
- xl (Extra Large) : Extra large size for emphasis and accessibility
- xxl (Ultra Large) : Maximum size for hero sections and special emphasis

Avatar Shapes

Two shape variants to match different design systems and preferences.

```

<lt;aava-avatars
  size="xxs"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="Ultra Small avatar"
<gt;
<lt;/aava-avatars>
<lt;aava-avatars
  size="sm"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="Small avatar"
<gt;
<lt;/aava-avatars>
<lt;aava-avatars
  size="md"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="Medium avatar"
<gt;
<lt;/aava-avatars>
<lt;aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="Large avatar"
<gt;
<lt;/aava-avatars>
<lt;aava-avatars
  size="xl"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="Extra Large avatar"
<gt;
<lt;/aava-avatars>
<lt;aava-avatars
  size="xxl"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="Ultra Large avatar"
<gt;
<lt;/aava-avatars>

```

Shape Features

- Pill : Circular shape with smooth rounded corners
- Square : Modern square shape with subtle border radius
- Responsive : Border radius adapts to size for optimal appearance

Badge Integration

Seamless integration with badge components for notifications and status indicators.

```
<aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  altText="Pill shape avatar"
>
</aava-avatars>
<aava-avatars
  size="lg"
  shape="square"
  imageUrl="assets/1.svg"
  altText="Square shape avatar"
>
</aava-avatars>
```

Badge Features

- Status Badges : Visual indicators for online/offline status
- Count Badges : Numeric indicators for notifications
- Custom States : Support for various badge states and colors
- Positioning : Automatic positioning at top-right corner
- Responsive : Badge size adapts to avatar size

Text Labels

Support for both status and profile text labels with flexible positioning.

```

<lt;aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  badgeState="high-priority"
  badgeSize="small"
  [badgeCount]="1"
  altText="Avatar with status badge"
<gt;
<lt;/aava-avatars>
<lt;aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  badgeState="information"
  badgeSize="small"
  [badgeCount]="5"
  altText="Avatar with count badge"
<gt;
<lt;/aava-avatars>
<lt;aava-avatars
  size="lg"
  shape="square"
  imageUrl="assets/1.svg"
  badgeState="medium-priority"
  badgeSize="small"
  [badgeCount]="12"
  altText="Avatar with notification badge"
<gt;
<lt;/aava-avatars>

```

Text Label Features

- Status Text : Small text for status indicators
- Profile Text : Larger text for user names or titles
- Dual Support : Can display both status and profile text simultaneously
- Typography : Different font sizes for hierarchy
- Alignment : Proper alignment with avatar element

Avatar Initials

Avatar can also support text-based initials

```

<aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  statusText="Online"
  altText="Avatar with status text"
</aava-avatars>
<aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  profileText="Marie Jane"
  altText="Avatar with profile text"
</aava-avatars>
<aava-avatars
  size="lg"
  shape="pill"
  imageUrl="assets/1.svg"
  statusText="Online"
  profileText="Marie Jane"
  altText="Avatar with both status and profile text"
</aava-avatars>

```

Initials Features

- Custom Initials : Allow developers to provide initials directly
- Background Colors : Configurable solid backgrounds
- Text Styling : Adjustable font size, weight, and color for initials visibility.
- Size Variants : Support for sm, md, lg (and more) to fit different UI contexts.

Accessibility Features

- Screen Reader Support : Proper ARIA labels and descriptions
- Focus Indicators : Clear visual focus indicators
- High Contrast : Enhanced visibility in high contrast mode
- Reduced Motion : Respects user motion preferences
- Alt Text : Support for image alternative text

API Reference

Inputs

Property	Type	Description
size	'xxs' 'xs' 'sm' 'md' 'lg' 'xl' 'xxl'	Size of the avatar component
shape	'pill' 'square'	Shape variant of the avatar
imageUrl	string	URL of the avatar image
statusText	string	Small text for status indicators
profileText	string	Larger text for user names or titles
badgeState	BadgeState	State of the badge component
badgeSize	BadgeSize	Size of the badge component
badgeCount	number	Numeric value for count badges
additionalText	string	Additional text label for extra information
initials	string	Initials shown in avatar
initialsBackground	string	Background color for initials
initialsColor	string	Text color for initials
customStyles	Record<string, string>	CSS custom properties override

CSS Custom Properties

Property	Default	Description
--avatar-size-us	Dynamic	Ultra small avatar size
--avatar-size-xs	Dynamic	Extra small avatar size
--avatar-size-sm	Dynamic	Small avatar size

Property	Default	Description
--avatar-size-md	Dynamic	Medium avatar size
--avatar-size-lg	Dynamic	Large avatar size
--avatar-size-xl	Dynamic	Extra large avatar size
--avatar-size-ul	Dynamic	Ultra large avatar size
--avatar-border-radius	Dynamic	Border radius for avatar shapes
--avatar-border-radius-us	Dynamic	Border radius for ultra small size
--avatar-border-radius-xs	Dynamic	Border radius for extra small size
--avatar-border-radius-sm	Dynamic	Border radius for small size
--avatar-border-radius-md	Dynamic	Border radius for medium size
--avatar-border-radius-lg	Dynamic	Border radius for large size
--avatar-border-radius-xl	Dynamic	Border radius for extra large size
--avatar-border-radius-ul	Dynamic	Border radius for ultra large size

Accessibility Guidelines

Screen Reader Support

- Provide meaningful alt text for avatar images
- Use descriptive labels for avatar purposes
- Include context about user status and information

Visual Design

- Ensure avatar images meet minimum size requirements
- Provide clear visual distinction between different states
- Support high contrast and reduced motion preferences

- Use consistent sizing and spacing across the interface

Best Practices

Design Guidelines

- Appropriate Sizing : Choose sizes based on context and importance
Ultra Small/Extra Small : Use for extremely dense layouts, data tables, and compact interfaces
Small/Medium : Use for standard layouts, navigation, and user lists
Large : Use for featured content, profile headers, and important user information
Extra Large/Ultra Large : Use for hero sections, prominent displays, and accessibility emphasis
- Consistent Shapes : Use consistent shapes within the same interface
- Clear Hierarchy : Use size and styling to indicate importance
- Status Clarity : Make status indicators clear and unambiguous
- Image Quality : Use high-quality, properly sized images

Performance

- Image Optimization : Optimize avatar images for web delivery
- Lazy Loading : Implement lazy loading for avatar images
- Caching : Cache frequently used avatar images
- Animation Performance : Use hardware-accelerated animations
- Bundle Optimization : Import only needed avatar features

User Experience

- Loading States : Provide loading indicators for remote images
- Fallback Handling : Implement graceful fallbacks for missing images
- Consistent Behavior : Maintain consistent behavior across avatar types
- Responsive Design : Ensure avatars work well on all screen sizes
- Touch Targets : Ensure adequate touch target sizes for mobile

Implementation Considerations

- Image Management : Implement proper image loading and error handling
- State Management : Properly manage avatar states in your application
- Accessibility : Ensure all avatar features are accessible
- Theming : Use CSS custom properties for consistent theming
- Testing : Test avatars across different devices and browsers

Accessibility Implementation

- Semantic HTML : Use proper HTML structure for avatar elements
- ARIA Attributes : Implement appropriate ARIA labels and roles
- Focus Management : Ensure logical focus order and indicators
- Screen Reader Testing : Test with actual screen readers
- Keyboard Testing : Verify complete keyboard navigation flow